

#Abyan Ardiatama

#24060120140161

#Praktikum ke-2 Machine Learning

##Soal 1

###Evaluasi Algoritma menggunakan dataset Iris dengan 3 algoritma yang berbeda dengan modul

###Memuat dataset ke dalam google colab menggunakan pandas dengan memasukkan url dan nama atributnya

```
import pandas
#Memuat Dataset
url =
"https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.d
ata"
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width',
'class']
dataset = pandas.read_csv(url, names=names)
print(dataset)
```

	sepal-length	sepal-width	petal-length	petal-width	
class					
0	5.1	3.5	1.4	0.2	Iris-
setosa					
1	4.9	3.0	1.4	0.2	Iris-
setosa					
2	4.7	3.2	1.3	0.2	Iris-
setosa					
3	4.6	3.1	1.5	0.2	Iris-
setosa					
4	5.0	3.6	1.4	0.2	Iris-
setosa					
..	
...					
145	6.7	3.0	5.2	2.3	Iris-
virginica					
146	6.3	2.5	5.0	1.9	Iris-
virginica					
147	6.5	3.0	5.2	2.0	Iris-
virginica					
148	6.2	3.4	5.4	2.3	Iris-
virginica					
149	5.9	3.0	5.1	1.8	Iris-
virginica					

[150 rows x 5 columns]

###Membagi dataset menjadi 80% data training dan 20% data validasi

```
from sklearn import*
array = dataset.values
X = array[:,0:4]
Y = array[:,4]
validation_size = 0.20
seed = 7
X_train, X_validation, Y_train, Y_validation =
model_selection.train_test_split(X, Y, test_size=validation_size,
random_state=seed)
```

###Memvalidasi dataset pada 3 algoritma (Decision Tree,Stochastic Gradient Descent,dan Gaussian Process)

Test options and evaluation metric

```
seed = 7
scoring = 'accuracy'
```

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import SGDClassifier
from sklearn.gaussian_process import GaussianProcessClassifier
from sklearn import*
```

Spot Check Algorithms

```
models = []
models.append(('Decision Tree',DecisionTreeClassifier()))
models.append(('Stochastic Gradient Descent',SGDClassifier()))
models.append(('Gaussian Process',GaussianProcessClassifier()))
```

evaluate each model in turn

```
results = []
names = []

for name, model in models:
    kfold = model_selection.KFold(n_splits=10, shuffle=True,
random_state=seed)
    cv_results = model_selection.cross_val_score(model, X_train,
Y_train, cv=kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

```
Decision Tree: 0.958333 (0.076830)
Stochastic Gradient Descent: 0.800000 (0.197906)
Gaussian Process: 0.983333 (0.033333)
```

###Meguji keakuratan metode yang memiliki akurasi perkiraan tertinggi --> **Gaussian Process**

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

```

from sklearn.metrics import classification_report
# Make predictions on validation dataset
GaussianProcess=GaussianProcessClassifier()
GaussianProcess.fit(X_train, Y_train)
predictions = GaussianProcess.predict(X_validation)
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))

```

0.8666666666666667

```

[[ 7  0  0]
 [ 0 10  2]
 [ 0  2  9]]

```

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	7
Iris-versicolor	0.83	0.83	0.83	12
Iris-virginica	0.82	0.82	0.82	11
accuracy			0.87	30
macro avg	0.88	0.88	0.88	30
weighted avg	0.87	0.87	0.87	30

#Abyan Ardiatama

#24060120140161

#Praktikum ke-2 Machine Learning

##Soal 2

###Mencari dataset baru dan evaluasi menggunakan 3 algoritma yang ada di modul

###Memuat dataset ke dalam google colab menggunakan pandas dengan memasukkan url dan nama atributnya

```
import pandas
#Memuat Dataset
url =
"https://archive.ics.uci.edu/ml/machine-learning-databases/blood-
transfusion/transfusion.data"
#names = ['mpg', 'cylinders', 'displacement', 'horsepower', 'weight']
dataset = pandas.read_csv(url)

print(dataset.head(5))
```

	Recency (months)	Frequency (times)	Monetary (c.c. blood)	Time
0	2	50	12500	
98				
1	0	13	3250	
28				
2	1	16	4000	
35				
3	2	20	5000	
45				
4	1	24	6000	
77				

	whether he/she donated blood in March 2007
0	1
1	1
2	1
3	1
4	0

###Membagi dataset menjadi 80% data training dan 20% data validasi

```
from sklearn import*
array = dataset.values
X = array[:,0:4]
Y = array[:,4]
validation_size = 0.20
seed = 7
```

```

X_train, X_validation, Y_train, Y_validation =
model_selection.train_test_split(X, Y, test_size=validation_size,
random_state=seed)

# Test options and evaluation metric
seed = 7
scoring = 'accuracy'

###Memvalidasi dataset pada 5 algoritma (KNN,GaussianNB,dan SVM)

from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn import*
# Spot Check Algorithms
models = []
models.append(('KNN', KNeighborsClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))

# evaluate each model in turn
results = []
names = []

for name, model in models:
    kfold = model_selection.KFold(n_splits=5, shuffle=True,
random_state=seed)
    cv_results = model_selection.cross_val_score(model, X_train,
Y_train, cv=kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    error_score='raise'
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)

KNN: 0.757521 (0.021764)
NB: 0.769314 (0.049525)
SVM: 0.762619 (0.028621)

```

###Meguji keakuratan metode yang memiliki akurasi perkiraan tertinggi --> **GaussianNB**

```

from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
# Make predictions on validation dataset
NB = GaussianNB()
NB.fit(X_train, Y_train)
predictions = NB.predict(X_validation)
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))

```

0.7066666666666667

[[98 14]

[30 8]]

	precision	recall	f1-score	support
0	0.77	0.88	0.82	112
1	0.36	0.21	0.27	38
accuracy			0.71	150
macro avg	0.56	0.54	0.54	150
weighted avg	0.66	0.71	0.68	150